# class08

Isabella Ruud: PID A59016138

## Table of contents



### Load the data

Today we will practice applying our PCA and clustering methods from the last class on some breast cancer FNA data.

First, let's get the data into R

```
# Save your input data file into your Project directory
fna.data <- "WisconsinCancer.csv"

# Complete the following code to input the data and store as wisc.df
wisc.df <- read.csv(fna.data, row.names=1)

head(wisc.df)
```

```
         diagnosis radius_mean texture_mean perimeter_mean area_mean
842302           M       17.99        10.38         122.80    1001.0
842517           M       20.57        17.77         132.90    1326.0
84300903         M       19.69        21.25         130.00    1203.0
84348301         M       11.42        20.38          77.58     386.1
84358402         M       20.29        14.34         135.10    1297.0
843786           M       12.45        15.70          82.57     477.1
```

```
         smoothness_mean compactness_mean concavity_mean concave.points_mean
842302           0.11840          0.27760         0.3001             0.14710
842517           0.08474          0.07864         0.0869             0.07017
84300903         0.10960          0.15990         0.1974             0.12790
84348301         0.14250          0.28390         0.2414             0.10520
84358402         0.10030          0.13280         0.1980             0.10430
843786           0.12780          0.17000         0.1578             0.08089
         symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
842302          0.2419                0.07871    1.0950     0.9053        8.589
842517          0.1812                0.05667    0.5435     0.7339        3.398
84300903        0.2069                0.05999    0.7456     0.7869        4.585
84348301        0.2597                0.09744    0.4956     1.1560        3.445
84358402        0.1809                0.05883    0.7572     0.7813        5.438
843786          0.2087                0.07613    0.3345     0.8902        2.217
         area_se smoothness_se compactness_se concavity_se concave.points_se
842302    153.40      0.006399        0.04904      0.05373           0.01587
842517     74.08      0.005225        0.01308      0.01860           0.01340
84300903   94.03      0.006150        0.04006      0.03832           0.02058
84348301   27.23      0.009110        0.07458      0.05661           0.01867
84358402   94.44      0.011490        0.02461      0.05688           0.01885
843786     27.19      0.007510        0.03345      0.03672           0.01137
         symmetry_se fractal_dimension_se radius_worst texture_worst
842302       0.03003             0.006193        25.38         17.33
842517       0.01389             0.003532        24.99         23.41
84300903     0.02250             0.004571        23.57         25.53
84348301     0.05963             0.009208        14.91         26.50
84358402     0.01756             0.005115        22.54         16.67
843786       0.02165             0.005082        15.47         23.75
         perimeter_worst area_worst smoothness_worst compactness_worst
842302            184.60     2019.0           0.1622            0.6656
842517            158.80     1956.0           0.1238            0.1866
84300903          152.50     1709.0           0.1444            0.4245
84348301           98.87      567.7           0.2098            0.8663
84358402          152.20     1575.0           0.1374            0.2050
843786            103.40      741.6           0.1791            0.5249
         concavity_worst concave.points_worst symmetry_worst
842302            0.7119               0.2654         0.4601
842517            0.2416               0.1860         0.2750
84300903          0.4504               0.2430         0.3613
84348301          0.6869               0.2575         0.6638
84358402          0.4000               0.1625         0.2364
843786            0.5355               0.1741         0.3985
         fractal_dimension_worst
```

```
842302                  0.11890
842517                  0.08902
84300903                0.08758
84348301                0.17300
84358402                0.07678
843786                  0.12440
```

Q1. How many patients/samples are in this dataset?

```
nrow(wisc.df)
```

```
[1] 569
```

There are 569 samples in this dataset.

Q2. How many cancer/non-cancer samples samples are in the dataset?

```
table(wisc.df$diagnosis)
```

```
  B   M
357 212
```

The `table()` function is super useful for counting up the number of observations of each type.

There are 357 benign and 212 malignant patient samples.

How many columns/dimensions are in this dataset?

```
ncol(wisc.df)
```

```
[1] 31
```

There are 31 columns, or 30 columns once you remove the diagnosis column.

Q3. How many columns are suffixed with "_mean"? The `grep()` function can help with pattern matching here

```
length(grep("_mean", colnames(wisc.df)))
```

```
[1] 10
```

There are 10 columns that are suffixed with _mean.

### Cluster the dataset

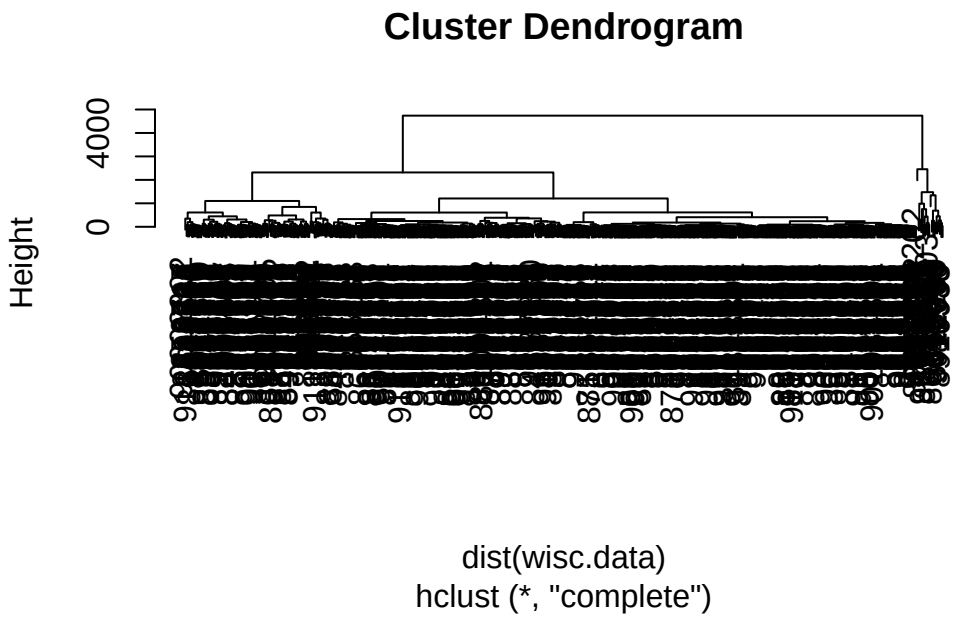
Let's try a `hclust()`

First, emove the diagnosis column (the first column) since we don't want to include it in the PCA or clustering

```
# We can use -1 here to remove the first column
wisc.data <- wisc.df[,-1]

# Create diagnosis vector for later
diagnosis <- wisc.df$diagnosis
```

```
hc.raw <- hclust(dist(wisc.data))
plot(hc.raw)
```




To get some clusters, I can use cutree() to cut the tree at a given height

```
grps <- cutree(hc.raw, h=4000)
table(grps)
```

```
grps
  1   2
549  20
```

To see the correspondance of our cluster groups `grps` with the expert `diagnosis` i can use table() again.

```
table(grps,diagnosis)
```

```
    diagnosis
grps   B   M
   1 357 192
   2   0  20
```

That is not a useful clustering result...

### Principal Component Analysis (PCA)

Scaling data before analysis is often critical.

Side note: the default for `prcomp()` is `scale=FALSE` There is a dataset in R called `mtcars` which has loads of numbers about old cars. The means and standard deviations in each column vary a lot from column to column. The disp and hp columns will dominate the analysis because they have high values

```
colMeans(mtcars)
```

```
       mpg        cyl       disp         hp       drat         wt       qsec
 20.090625   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750
        vs         am       gear       carb
  0.437500   0.406250   3.687500   2.812500
```

```
apply(mtcars, 2, sd)
```

```
       mpg        cyl       disp         hp       drat         wt
 6.0269481  1.7859216 123.9386938 68.5628685  0.5346787  0.9784574
      qsec         vs         am       gear       carb
 1.7869432  0.5040161  0.4989909  0.7378041  1.6152000
```

Let's see what the effect of scaling is on the PCA

```
pc.noscale <- prcomp(mtcars, scale=FALSE)
pc.scale <- prcomp(mtcars, scale=TRUE)
```

let's look at the loadings first

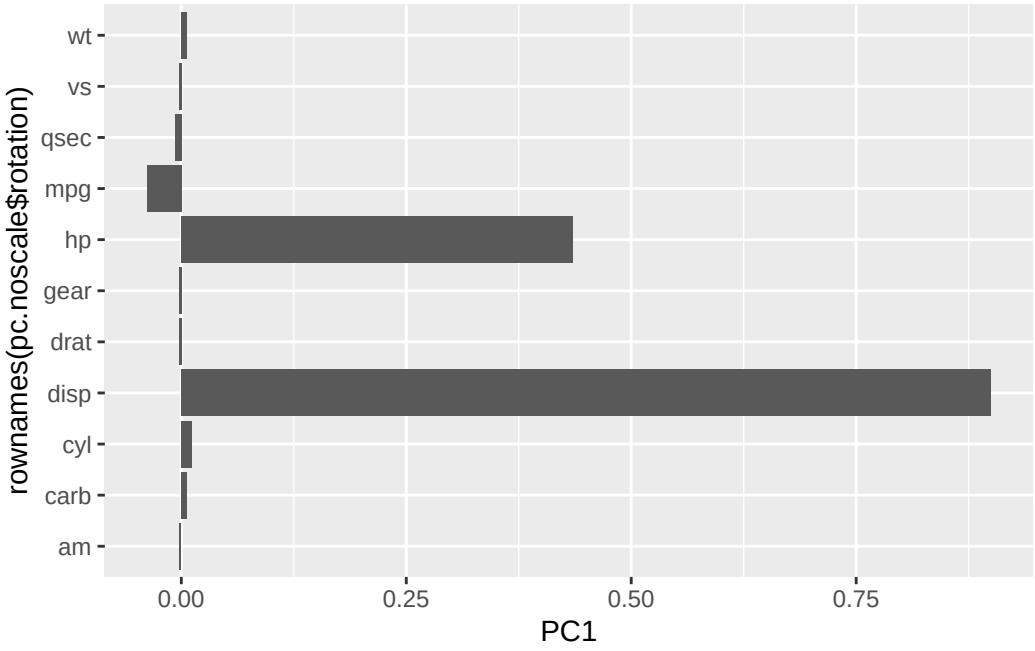
```
pc.noscale$rotation
```

```
              PC1          PC2          PC3          PC4         PC5
mpg  -0.038118199  0.009184847  0.982070847  0.047634784 -0.08832843
cyl   0.012035150 -0.003372487 -0.063483942 -0.227991962  0.23872590
disp  0.899568146  0.435372320  0.031442656 -0.005086826 -0.01073597
hp    0.434784387 -0.899307303  0.025093049  0.035715638  0.01655194
drat -0.002660077 -0.003900205  0.039724928 -0.057129357 -0.13332765
wt    0.006239405  0.004861023 -0.084910258  0.127962867 -0.24354296
qsec -0.006671270  0.025011743 -0.071670457  0.886472188 -0.21416101
vs   -0.002729474  0.002198425  0.004203328  0.177123945 -0.01688851
am   -0.001962644 -0.005793760  0.054806391 -0.135658793 -0.06270200
gear -0.002604768 -0.011272462  0.048524372 -0.129913811 -0.27616440
carb  0.005766010 -0.027779208 -0.102897231 -0.268931427 -0.85520810
              PC6          PC7           PC8          PC9          PC10
mpg  -0.143790084 -0.039239174 -2.271040e-02 -0.002790139  0.030630361
cyl  -0.793818050  0.425011021  1.890403e-01  0.042677206  0.131718534
disp  0.007424138  0.000582398  5.841464e-04  0.003532713 -0.005399132
hp    0.001653685 -0.002212538 -4.748087e-06 -0.003734085  0.001862554
drat  0.227229260  0.034847411  9.385817e-01 -0.014131110  0.184102094
wt   -0.127142296 -0.186558915 -1.561907e-01 -0.390600261  0.829886844
qsec -0.189564973  0.254844548  1.028515e-01 -0.095914479 -0.204240658
vs    0.102619063 -0.080788938  2.132903e-03  0.684043835  0.303060724
am    0.205217266  0.200858874  2.273255e-02 -0.572372433 -0.162808201
gear  0.334971103  0.801625551 -2.174878e-01  0.156118559  0.203540645
carb -0.283788381 -0.165474186 -3.972219e-03  0.127583043 -0.239954748
              PC11
mpg   0.0158569365
cyl  -0.1454453628
disp -0.0009420262
hp    0.0021526102
drat  0.0973818815
wt    0.0198581635
qsec -0.0110677880
vs   -0.6256900918
am   -0.7331658036
gear  0.1909325849
carb -0.0557957968
```

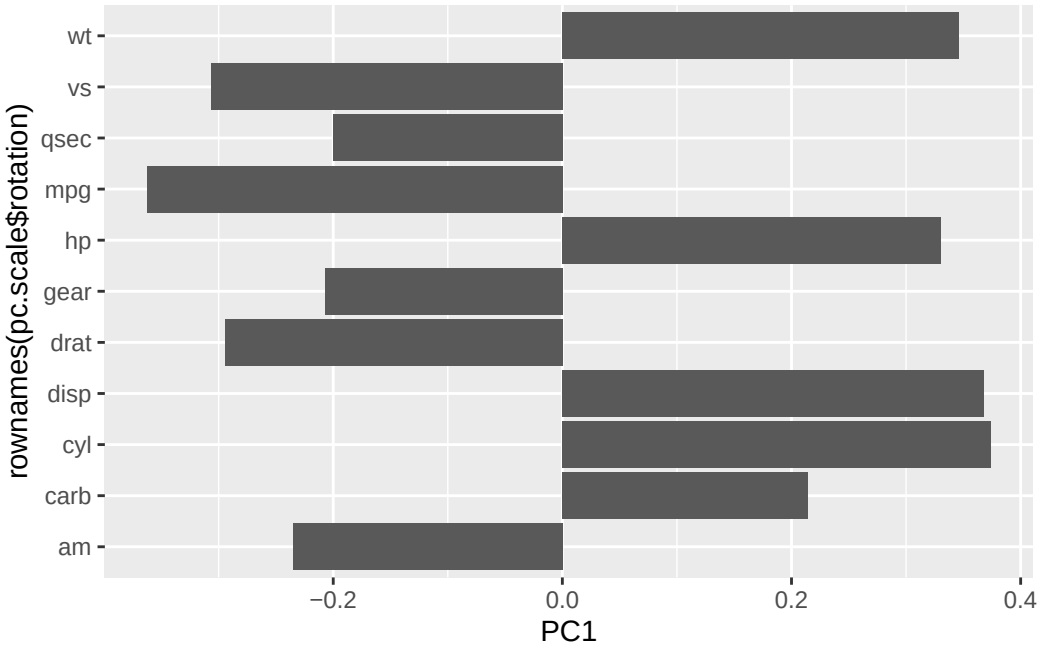
```
pc.scale$rotation
```

```
             PC1         PC2         PC3          PC4         PC5         PC6
mpg  -0.3625305  0.01612440 -0.22574419 -0.022540255 -0.10284468 -0.10879743
cyl   0.3739160  0.04374371 -0.17531118 -0.002591838 -0.05848381  0.16855369
disp  0.3681852 -0.04932413 -0.06148414  0.256607885 -0.39399530 -0.33616451
hp    0.3300569  0.24878402  0.14001476 -0.067676157 -0.54004744  0.07143563
drat -0.2941514  0.27469408  0.16118879  0.854828743 -0.07732727  0.24449705
wt    0.3461033 -0.14303825  0.34181851  0.245899314  0.07502912 -0.46493964
qsec -0.2004563 -0.46337482  0.40316904  0.068076532  0.16466591 -0.33048032
vs   -0.3065113 -0.23164699  0.42881517 -0.214848616 -0.59953955  0.19401702
am   -0.2349429  0.42941765 -0.20576657 -0.030462908 -0.08978128 -0.57081745
gear -0.2069162  0.46234863  0.28977993 -0.264690521 -0.04832960 -0.24356284
carb  0.2140177  0.41357106  0.52854459 -0.126789179  0.36131875  0.18352168
              PC7          PC8          PC9         PC10          PC11
mpg   0.367723810  0.754091423 -0.235701617 -0.13928524 -0.124895628
cyl   0.057277736  0.230824925 -0.054035270  0.84641949 -0.140695441
disp  0.214303077 -0.001142134 -0.198427848 -0.04937979  0.660606481
hp   -0.001495989  0.222358441  0.575830072 -0.24782351 -0.256492062
drat  0.021119857 -0.032193501  0.046901228  0.10149369 -0.039530246
wt   -0.020668302  0.008571929 -0.359498251 -0.09439426 -0.567448697
qsec  0.050010522  0.231840021  0.528377185  0.27067295  0.181361780
vs   -0.265780836 -0.025935128 -0.358582624  0.15903909  0.008414634
am   -0.587305101  0.059746952  0.047403982  0.17778541  0.029823537
gear  0.605097617 -0.336150240  0.001735039  0.21382515 -0.053507085
carb -0.174603192  0.395629107 -0.170640677 -0.07225950  0.319594676
```

```
library(ggplot2)
ggplot(pc.noscale$rotation) + aes(PC1, rownames(pc.noscale$rotation)) + geom_col()
```



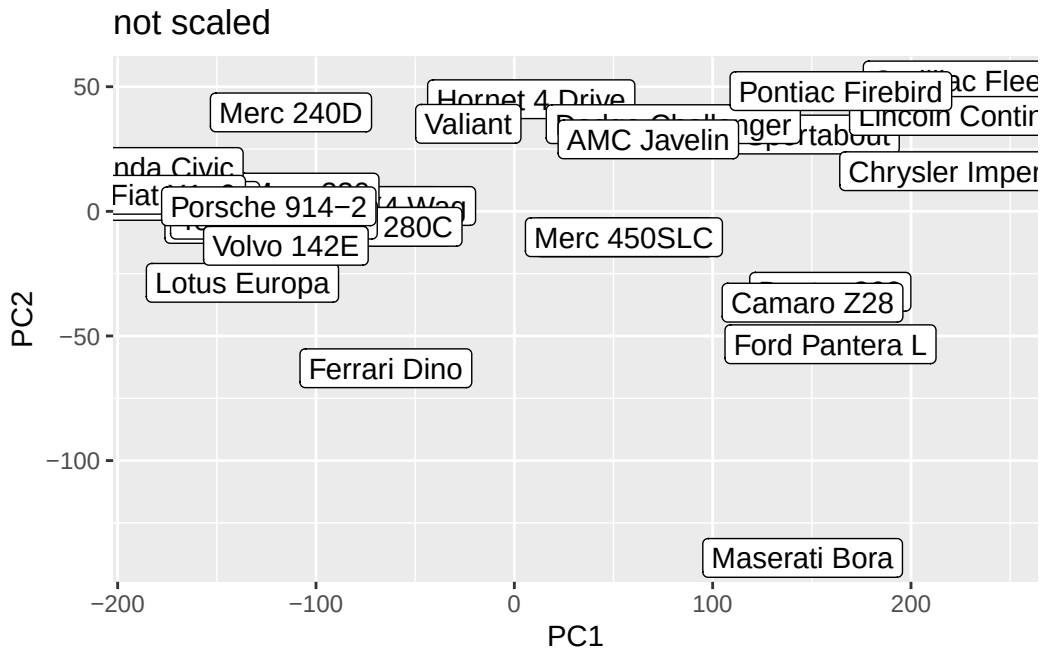

```
ggplot(pc.scale$rotation) + aes(PC1, rownames(pc.scale$rotation)) + geom_col()
```




The scaled one looks much more equal in distribution

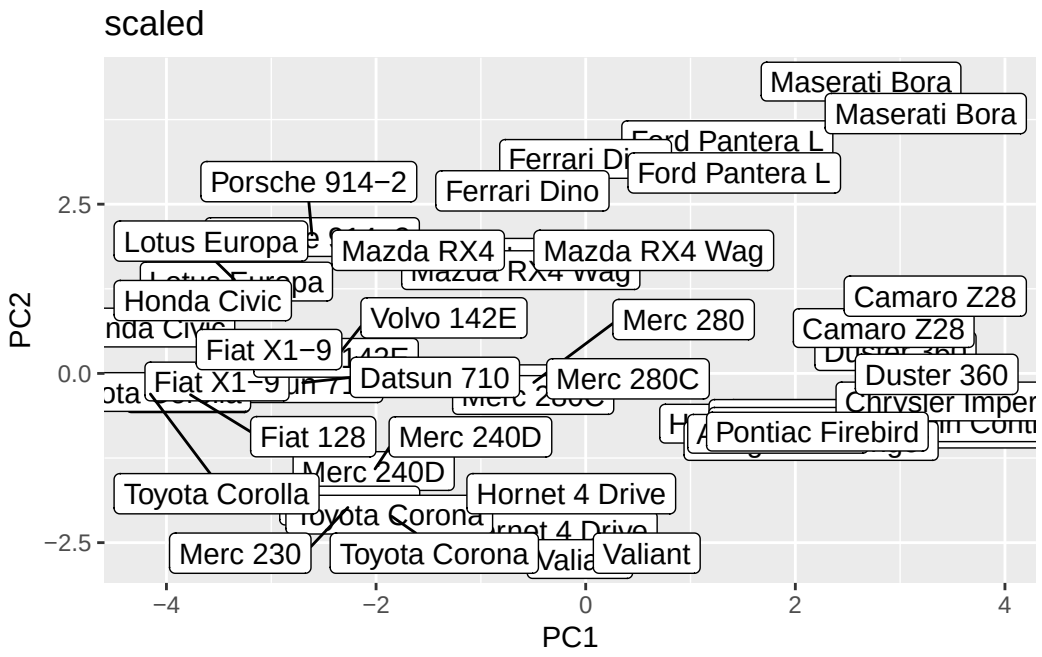
The main PC result figure is called a score plot or PC plot or PC1 vs PC2 plot

```
ggplot(pc.noscale$x) + aes(PC1, PC2, label = rownames(pc.noscale$x)) +
  geom_point() + geom_label() + labs(title="not scaled")
```




```
library(ggrepel)
ggplot(pc.scale$x) + aes(PC1, PC2, label = rownames(pc.scale$x)) +
  geom_point() + geom_label() + geom_label_repel() + labs(title = "scaled")
```

```
Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




The no scale PCA plot is dominated by horsepower, whereas there are more relationships shown in the scaled PCA

```
x <- scale(mtcars)
colMeans(x)
```

```
          mpg           cyl          disp            hp          drat
7.112366e-17 -1.474515e-17 -9.085614e-17  1.040834e-17 -2.918672e-16
           wt          qsec            vs            am          gear
4.682398e-17  5.299580e-16  6.938894e-18  4.510281e-17 -3.469447e-18
         carb
3.165870e-17
```

```
round(colMeans(x))
```

```
 mpg  cyl disp   hp drat   wt qsec   vs   am gear carb
   0    0    0    0    0    0    0    0    0    0    0
```

```
round(apply(x,2,sd))
```

```
 mpg  cyl disp   hp drat   wt qsec   vs   am gear carb
   1    1    1    1    1    1    1    1    1    1    1
```

**Key point**: generally we want to scale our data before analysis to avoid being misled due to your data having different measurement units

### Breast Cancer PCA

We will scale our data. Can check the means and standard deviations of the columns to see if they are different to determine if we need to scale.

```
pca <- prcomp(wisc.data, scale=TRUE)
```

Q4. What proportion of the original variance is captured by PC1?

PC1 captures 44% of the variance

Q5. How many principal components capture 70% of the variance?

You need 3 principal components to capture at least 70% of the variance.

Q6. How many principal components capture 90% of the variance?

You need 7 principal components to capture at least 90% of the variance.

See how well we are doing:

```
summary(pca)
```

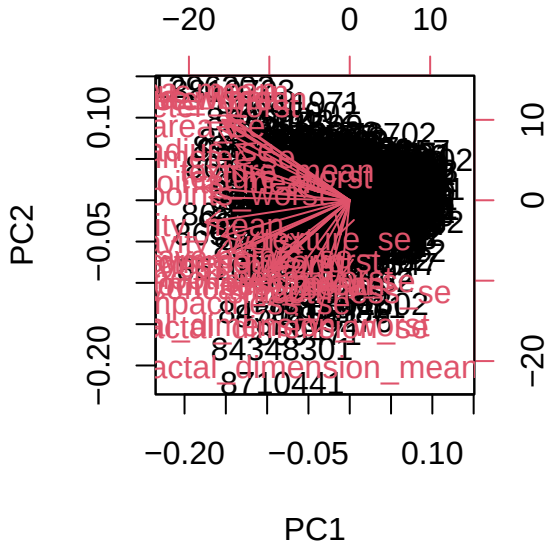
```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
                          PC15    PC16    PC17    PC18    PC19    PC20   PC21
Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
```

```
                            PC29    PC30
Standard deviation     0.02736 0.01153
Proportion of Variance 0.00002 0.00000
Cumulative Proportion  1.00000 1.00000
```

Q7. What stands out about this plot? Is it easy to understand?

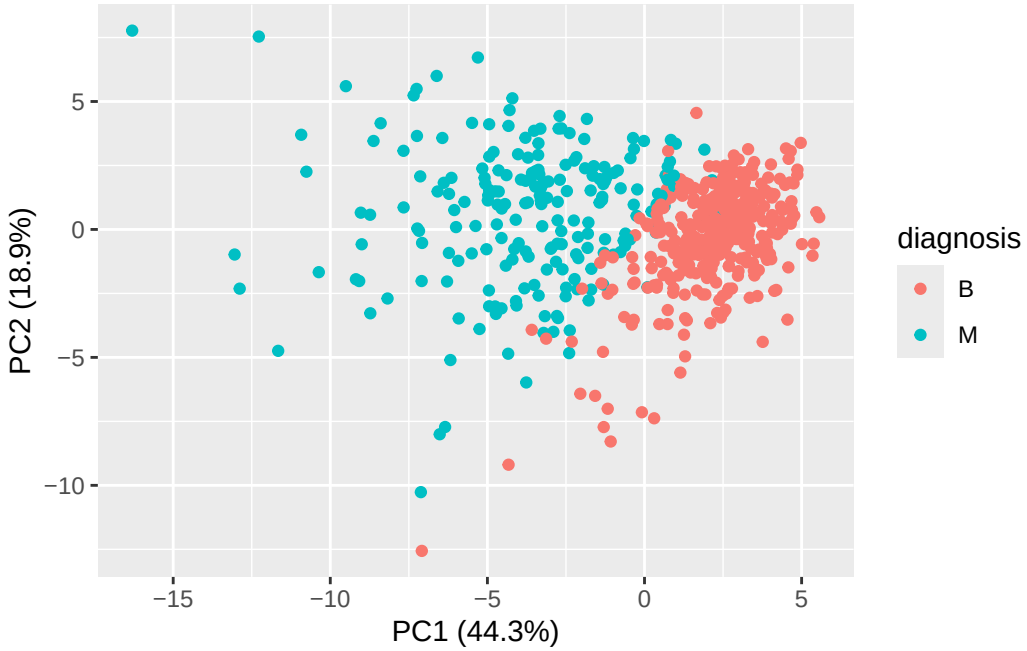
This plot is very difficult to interpet since there are numbers and arrows in one big mess. We will have to use other plotting methods to more clearly understand the PCA model.

```
biplot(pca)
```




Our PC plot

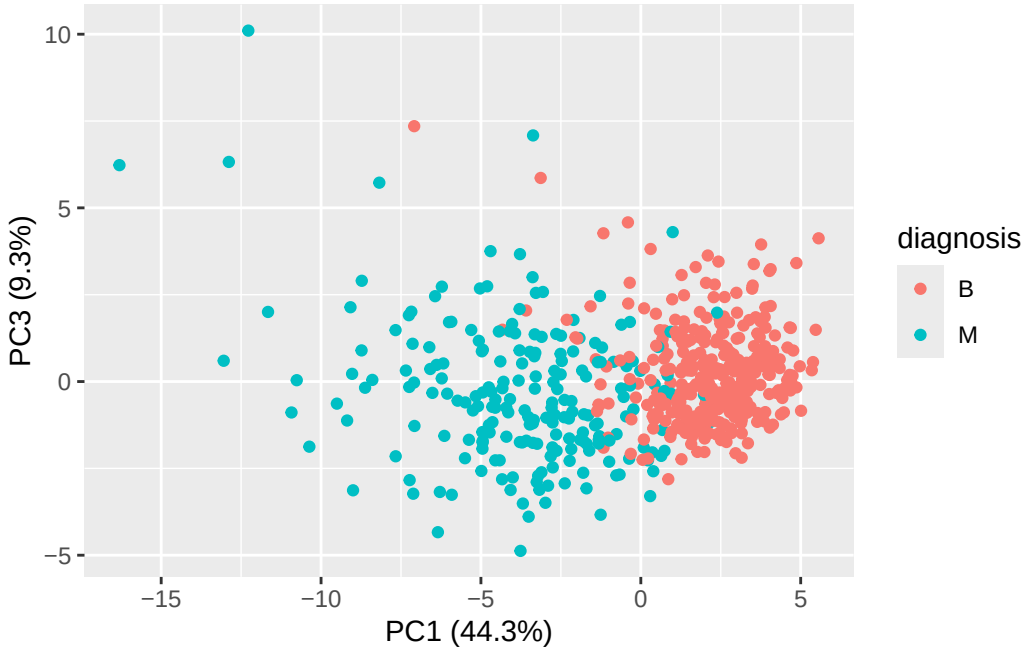
```
ggplot(pca$x) +
  aes(PC1, PC2, col = diagnosis) +
  geom_point() +
  xlab("PC1 (44.3%)") +
  ylab("PC2 (18.9%)")
```




Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

There is a similar grouping between malignant and benign, but the boundaries between the groups are a little less clear and there is some mixing of blue and red points where the groups meet.

```
ggplot(pca$x) +
  aes(PC1, PC3, col = diagnosis) +
  geom_point() +
  xlab("PC1 (44.3%)") +
  ylab("PC3 (9.3%)")
```




Q9. For the first principal component, what is the component of the loading vector (i.e. wisc.pr$rotation[,1]) for the feature concave.points_mean? This tells us how much this original feature contributes to the first PC.

The concave.points_mean is -0.26085376 for the the first PC.

```
pca$rotation[,1]
```

```
            radius_mean            texture_mean          perimeter_mean
             -0.21890244              -0.10372458              -0.22753729
              area_mean         smoothness_mean        compactness_mean
             -0.22099499              -0.14258969              -0.23928535
         concavity_mean     concave.points_mean           symmetry_mean
             -0.25840048              -0.26085376              -0.13816696
fractal_dimension_mean               radius_se              texture_se
             -0.06436335              -0.20597878              -0.01742803
           perimeter_se                 area_se           smoothness_se
             -0.21132592              -0.20286964              -0.01453145
        compactness_se           concavity_se      concave.points_se
             -0.17039345              -0.15358979              -0.18341740
          symmetry_se     fractal_dimension_se            radius_worst
             -0.04249842              -0.10256832              -0.22799663
         texture_worst          perimeter_worst               area_worst
```

```
          -0.10446933             -0.23663968             -0.22487053
     smoothness_worst       compactness_worst         concavity_worst
          -0.12795256             -0.21009588             -0.22876753
concave.points_worst          symmetry_worst fractal_dimension_worst
          -0.25088597             -0.12290456             -0.13178394
```

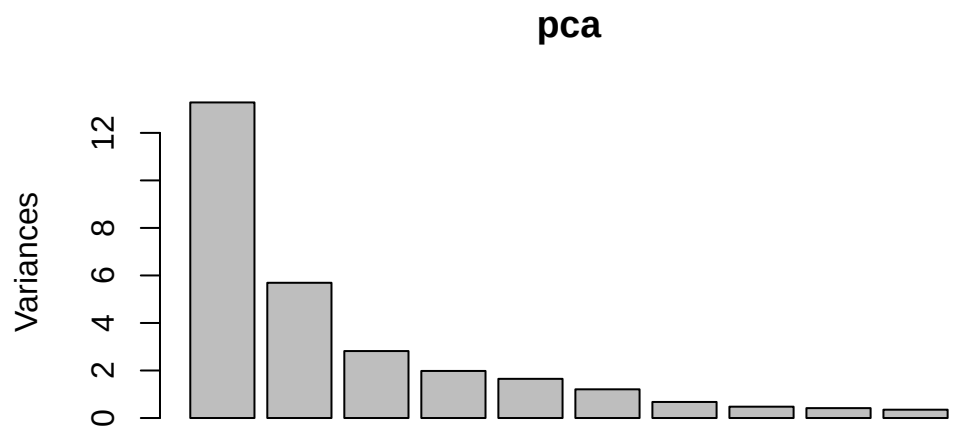
How many PCs capture 80% of the original variance in the dataset?

```
summary(pca)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
                          PC15    PC16    PC17    PC18    PC19    PC20   PC21
Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
                          PC29    PC30
Standard deviation     0.02736 0.01153
Proportion of Variance 0.00002 0.00000
Cumulative Proportion  1.00000 1.00000
```

From the summary, we can see that it takes 4 PCs to cover 79% of the variance, or 5 PCs to cover 84% of the variance

```
plot(pca)
```




Use ggplot to plot a scree plot of the variance per PC

```
attributes(pca)
```

```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

We can extract the sdev and square it to figure out the variance. the sum of the variance should be 30 because we scaled our data and there are 30 columns

```
v <- pca$sdev^2
sum(v)
```

```
[1] 30
```

The proportion of variance captured in each PC

```r
round(v/sum(v),2)
```

```
 [1] 0.44 0.19 0.09 0.07 0.05 0.04 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.00
[16] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

Cumulative variance captured

```r
cumsum(v/sum(v))
```

```
 [1] 0.4427203 0.6324321 0.7263637 0.7923851 0.8473427 0.8875880 0.9100953
 [8] 0.9259825 0.9398790 0.9515688 0.9613660 0.9700714 0.9781166 0.9833503
[15] 0.9864881 0.9891502 0.9911302 0.9928841 0.9945334 0.9955720 0.9965711
[22] 0.9974858 0.9982971 0.9988990 0.9994150 0.9996876 0.9999176 0.9999706
[29] 0.9999956 1.0000000
```

How many PCs capture 80% of the variance?

```r
cumsum(v/sum(v)) > 0.8
```

```
 [1] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

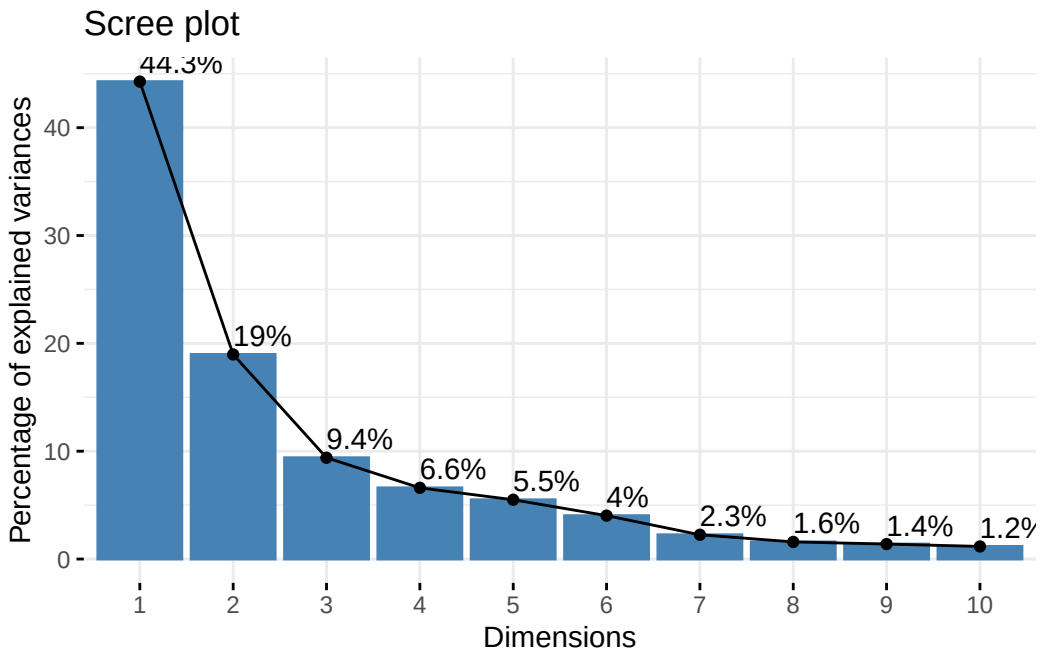
```r
which(cumsum(v/sum(v)) > 0.8)
```

```
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
[26] 30
```

```r
#install.packages("factoextra")
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
fviz_eig(pca, addlabels = TRUE)
```

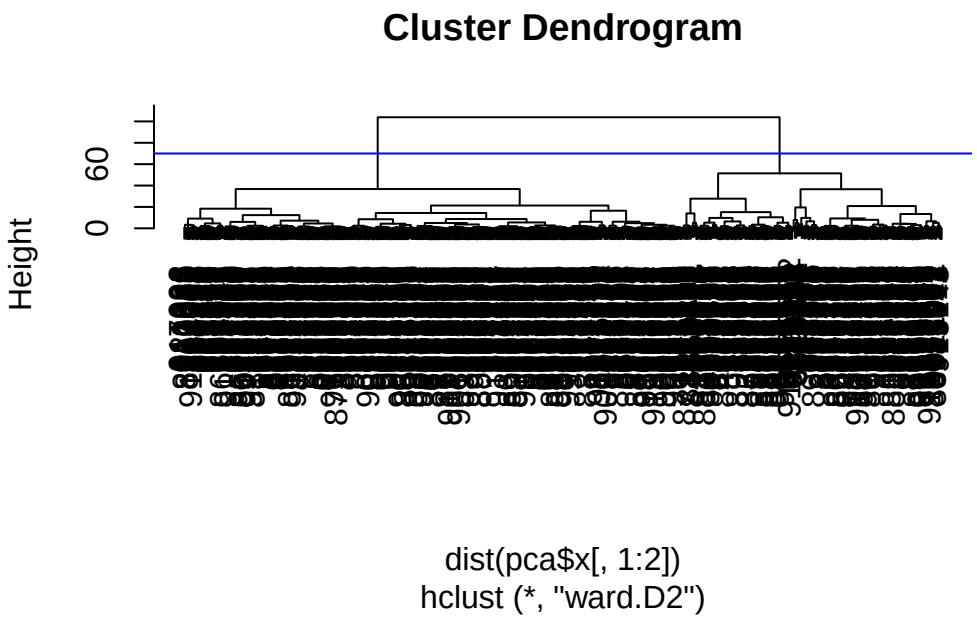



### Combine PCA and clustering

We saw earlier that clustering the raw data alone did not provide useful results

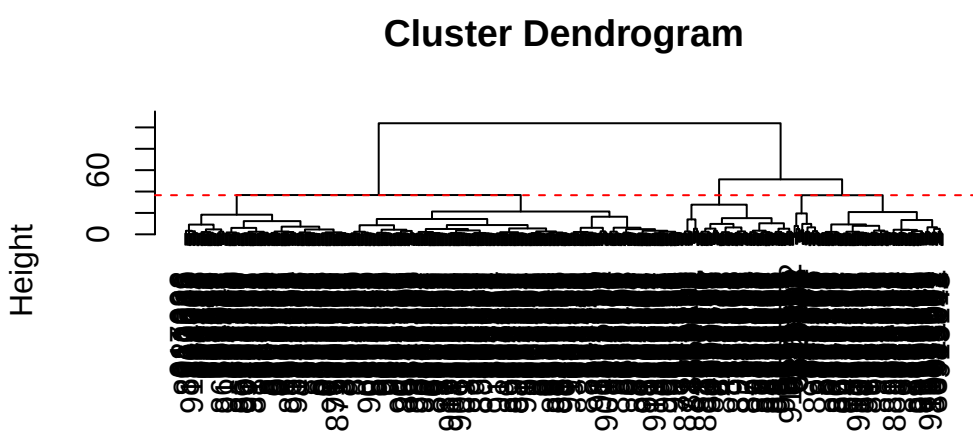
We can use our new PC variables (our PCs) as a basis for clustering. Use $x (PC scores) and use some of the PCs to do the clustering (if you use all of them, then it's like the clustering from before). We will cluster in the PC1 and PC2 subspace

```
hc.pca <- hclust(dist(pca$x[,1:2]), method="ward.D2")
plot(hc.pca)
abline(h=70, col="blue")
```




Q10. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

```
plot(hc.pca)
abline(h=36.6, col="red", lty=2)
```



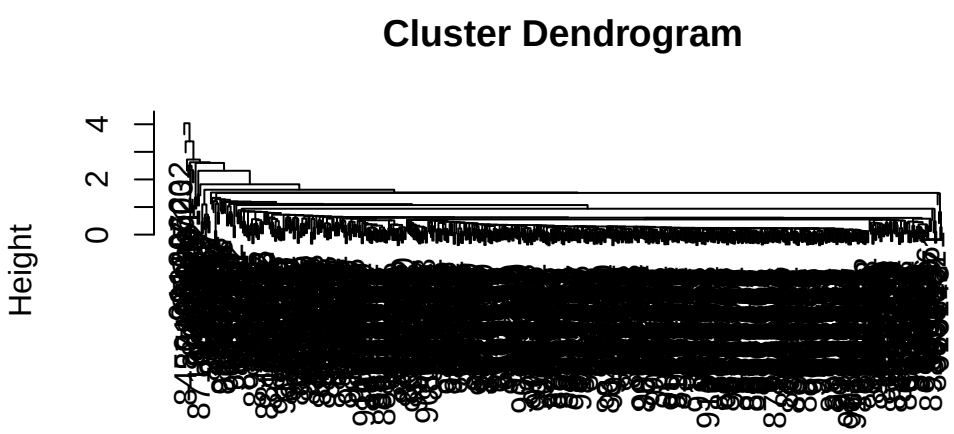

```
wisc.hclust.clusters <- cutree(hc.pca, h=36.6)
table(wisc.hclust.clusters, diagnosis)
```

```
                    diagnosis
wisc.hclust.clusters   B   M
                   1   0 112
                   2  18  65
                   3 232  18
                   4 107  17
```

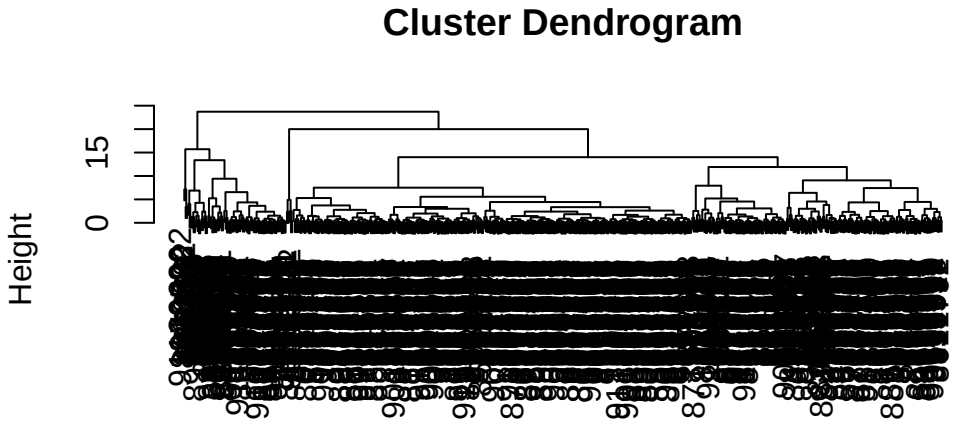
Q12. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

ward.D2 is my favorite method because it is the simplest tree and starts by splitting in half into two relatively even branches. The other methods very quickly separate into many branches.

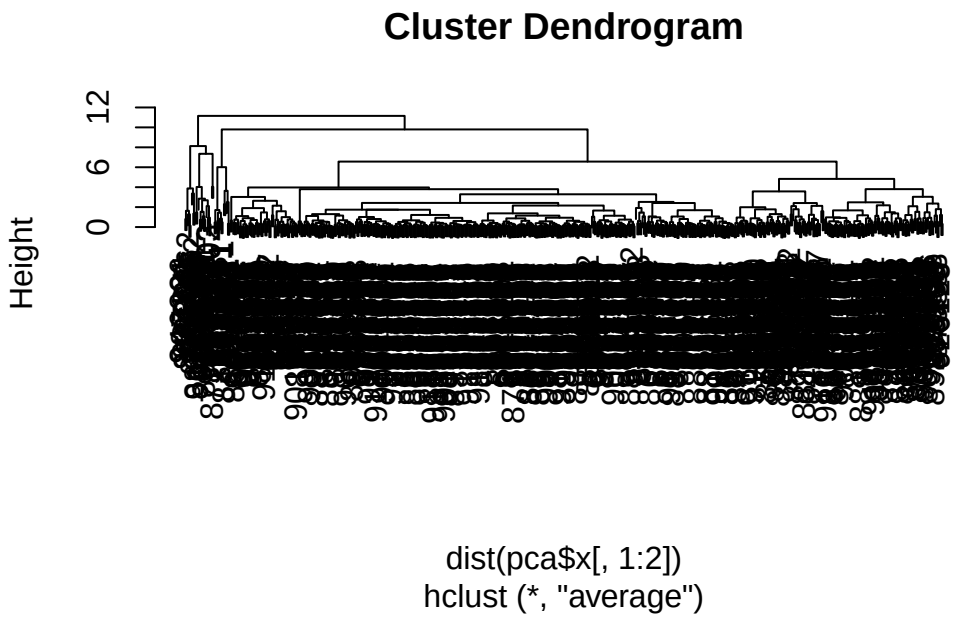
```
hc.pca <- hclust(dist(pca$x[,1:2]), method="single")
plot(hc.pca)
```




```
hc.pca <- hclust(dist(pca$x[,1:2]), method="complete")
plot(hc.pca)
```



```
hc.pca <- hclust(dist(pca$x[,1:2]), method="average")
plot(hc.pca)
```




Q13. How well does the newly created model with four clusters separate out the two diagnoses?

Does your clustering help separate cancer from non-cancer? (ie diagnosis M vs B)

```
grps <- cutree(hc.pca, h=70)
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1 357 212
```

Yes, group 1 is mostly M (cancer) and group 2 is mostly B (benign), although the clustering is not perfect

Q15. OPTIONAL: Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

Positive cancer samples “M” Negative non-cancer samples “B”

True our cluster/group 1 False our cluster/group 2

How many true positives do we have? 177 true positives

How many false positives do we have? 35 false positives

Sensitivity: true positives/(true positives + false negatives) $177/(177 + 339) = 0.52$ Specificity: true positives/(true negatives + false negatives) $177(18 + 339) = 0.5$

> Q14. How well do the hierarchical clustering models you created in previous sections (i.e. before PCA) do in terms of separating the diagnoses? Again, use the table() function to compare the output of each model (wisc.km$cluster and wisc.hclust.clusters) with the vector containing the actual diagnoses.

Without using the PCA model, the clustering is not very good since most of the benign and malignant diagnoses fall into the same group. Once you use the PCA in addition to the hierarchical clustering, the malignant and benign diagnoses start to separate into 2 different groups.

```
#clustering from before the PCA model
grps <- cutree(hc.raw, h=4000)
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1 357 192
   2   0  20
```

### Prediction with our PCA model

> Q16. Which of these new patients should we prioritize for follow up based on your results?

We can take new data (in this case from UofM) and project it onto our new variables (PCs)

load data

```
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
```

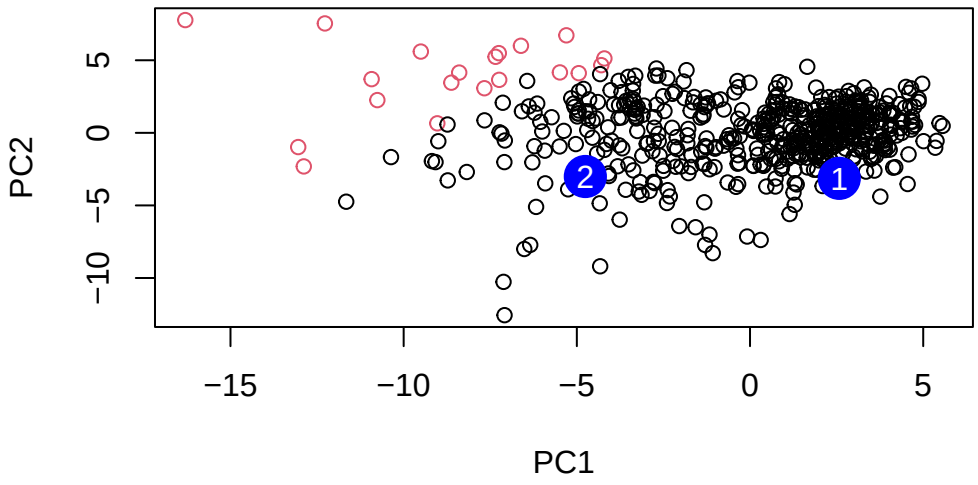
projection

```
npc <- predict(pca, newdata=new)
```

Base R plot

```
plot(pca$x[,1:2], col=grps)

#add the new points
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```




Patient 2 looks like they could have a malignant sample, so we should follow up with them